CLAIMS

What is claimed is:

1. A method for determining complexity of an enterprise information resource management system, the enterprise information resource management system being used to contain an ontology into which a plurality of enterprise data assets are mapped, the ontology including a plurality of model constructs, the enterprise data assets including a plurality of assets constructs, and the mappings between the data assets and the ontology including a plurality of mapping constructs, comprising:

receiving (i) a quantity of distinct asset constructs, denoted by C_{ASSET}, (ii) a quantity of distinct mapping constructs, denoted by $C_{MAPPING}$, and (iii) a quantity of distinct model constructs, denoted by C_{MODEL};

evaluating a metric of complexity, denoted by M, for an enterprise information resource management system having a capacity corresponding to C_{ASSET}, $C_{MAPPING}$ and C_{MODEL}, according to a formula

$$M = f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X),$$

where f is a real-valued function of three or more real-valued parameters and X denotes optional additional parameters; and

using the metric M within a transaction processing system, for license of the enterprise information resource management system.

2. The method of claim 1 wherein the plurality of data assets include assets that conform to a general data schema that uses element group asset constructs and element asset constructs.

3. The method of claim 2 wherein the general data schema is a relational database schema, and the element group asset constructs are database tables and the element asset constructs are columns of database tables.

4. The method of claim 2 wherein the general data schema is an XML schema, and the element group asset constructs are XML complex types and the element asset constructs are XML elements.

5. The method of claim 2 wherein the general data schema is a Cobol Copy Book, and the element group asset constructs are Cobol group items and the element asset constructs are Cobol elementary items.

6. The method of claim 2 wherein the ontology is an ontology model, and wherein the model constructs include ontology classes and their properties, and business rules that inter-relate the properties.

7. The method of claim 6 wherein the mapping constructs include mappings of element group asset constructs into ontology classes and mapping of element asset constructs into ontology properties.

8. The method of claim 6 wherein the function $f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X)$ is a step function of the form

$$f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X) = f_n(X), \text{ if } C_{n-1} < C \leq C_n,$$

where C is the total number of constructs, $C = C_{ASSET} + C_{MAPPING} + C_{MODEL}$, and where $C_0, C_1, C_2, \ldots$ are cutoff points.

9. The method of claim 6 wherein the function $f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X)$ is a step function of the form

$$f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X) = f_n(X), \text{ if } C_{n-1} < C \leq C_n,$$

where C is a weighted average $C = W_{ASSET} C_{ASSET} + W_{MAPPING} C_{MAPPING} + W_{MODEL} C_{MODEL}$, and where W_{ASSET}, $W_{MAPPING}$ and W_{MODEL} are respective weighting factors, and where $C_0, C_1, C_2, \ldots$ are cutoff points.

10. The method of claim 1 wherein the enterprise information resource management system generates results for tasks, and wherein the metric M also depends on the number of distinct results generated and saved.

11. The method of claim 10 wherein the results include data transformations.

12. The method of claim 10 wherein the results include SQL queries.

13. The method of claim 10 wherein the results include XSLT scripts.

14. The method of claim 1 wherein the enterprise information resource management system generates scripts for producing reports, and wherein the metric M also depends on the number of distinct report scripts generated and saved.

15. The method of claim 1 wherein the enterprise information resource management system records metadata, and wherein the metric M also depends on the number of distinct metadata records.

16. A method for determining complexity of a metadata repository including a plurality of metadata constructs, comprising:

receiving a quantity of distinct metadata constructs, denoted by C;

evaluating a metric of complexity, denoted by M, for a metadata repository having a capacity corresponding to C, according to a formula

$$M = f(C, X),$$

where f is a real-valued function of one or more real-valued parameters and X denotes optional additional parameters; and

using the metric M within a transaction processing system, for license of the metadata repository.

17. The method of claim 16 wherein metadata constructs are instances of meta-model constructs.

18. The method of claim 17 wherein the meta-data constructs are constructs for meta-models of asset schemas, and wherein the price P also depends on the number of meta-model constructs.

19. The method of claim 17 wherein at least one meta-model is a schema for relational database schemas, and wherein at least one metadata construct corresponds to a table of a relational database schema.

20. The method of claim 17 wherein at least one meta-model is a schema for XML schemas, and wherein at least one metadata construct corresponds to a complex element of an XML schema.

21. A method for determining complexity of a metadata repository including a plurality of metadata constructs, the metadata constructs being instances of meta-model constructs for meta-models of schemas for data assets, comprising:

receiving a quantity of distinct meta-model constructs, denoted by C;

evaluating a metric of complexity, denoted by M, for a metadata repository having a capacity corresponding to C, according to a formula

$$M = f(C, X),$$

where f is a real-valued function of one or more real-valued parameters and X denotes optional additional parameters; and

using the metric M within a transaction processing system, for license of the metadata repository.

22. The method of claim 21 wherein at least one meta-model is a schema for relational database schemas, and wherein at least one metadata construct corresponds to a table of a relational database schema.

23. The method of claim 21 wherein at least one meta-model is a schema for XML schemas, and wherein at least one metadata construct corresponds to a complex element of an XML schema.

24. The method of claim 21 wherein the meta-model constructs include descriptors for data assets.

25. The method of claim 21 wherein the optional additional parameters include a parameter for a number of users of the repository.

26. The method of claim 21 wherein the optional additional parameters include a parameter for a number of available features.

27. The method of claim 26 wherein an available feature is the ability to change a meta-model.

28. The method of claim 26 wherein an available feature is the ability to batch scan metadata.

29. A system for determining complexity of an enterprise information resource management system, the enterprise information resource management system being used to contain an ontology into which a plurality of enterprise data assets are mapped, the ontology including a plurality of model constructs, the enterprise data assets including a plurality of assets constructs, and the mappings between the data assets and the ontology including a plurality of mapping constructs, comprising:

an input device for receiving (i) a quantity of distinct asset constructs, denoted by C_{ASSET}, (ii) a quantity of distinct mapping constructs, denoted by $C_{MAPPING}$, and (iii) a quantity of distinct model constructs, denoted by C_{MODEL};

a processor coupled to said input device for evaluating a metric of complexity, denoted by M, for an enterprise information resource management system with capacity corresponding to C_{ASSET}, $C_{MAPPING}$ and C_{MODEL}, according to a formula

$$M = f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X),$$

where f is a real-valued function of three or more real-valued parameters and X denotes optional additional parameters; and

a transaction processing system receiving the metric M for licensing the enterprise information resource management system.

30. The system of claim 29 wherein the plurality of data assets include assets that conform to a general data schema that uses element group asset constructs and element asset constructs.

31. The system of claim 30 wherein the general data schema is a relational database schema, and the element group asset constructs are database tables and the element asset constructs are columns of database tables.

32. The system of claim 30 wherein the general data schema is an XML schema, and the element group asset constructs are XML complex types and the element asset constructs are XML elements.

33. The system of claim 30 wherein the general data schema is a Cobol Copy Book, and the element group asset constructs are Cobol group items and the element asset constructs are Cobol elementary items.

34. The system of claim 30 wherein the ontology is an ontology model, and wherein the model constructs include ontology classes and their properties, and business rules that inter-relate the properties.

35. The system of claim 34 wherein the mapping constructs include mappings of element group asset constructs into ontology classes and mapping of element asset constructs into ontology properties.

36. The system of claim 34 wherein the function $f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X)$ is a step function of the form

$$f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X) = f_n(X), \text{ if } C_{n-1} < C \leq C_n,$$

where C is the total number of constructs, $C = C_{ASSET} + C_{MAPPING} + C_{MODEL}$, and where C_0, C_1, C_2, ... are cutoff points.

37. The system of claim 34 wherein the function $f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X)$ is a step function of the form

$$f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X) = f_n(X), \text{ if } C_{n-1} < C \leq C_n,$$

where C is a weighted average $C = W_{ASSET} C_{ASSET} + W_{MAPPING} C_{MAPPING} + W_{MODEL} C_{MODEL}$, and where W_{ASSET}, $W_{MAPPING}$ and W_{MODEL} are respective weighting factors, and where C_0, C_1, C_2, ... are cutoff points.

38. The system of claim 29 wherein the enterprise information resource management system generates results for tasks, and wherein the metric M also depends on the number of distinct results generated and saved.

39. The system of claim 38 wherein the results include data transformations.

40. The system of claim 38 wherein the results include SQL queries.

41. The system of claim 38 wherein the results include XSLT scripts.

42. The system of claim 29 wherein the enterprise information resource management system generates scripts for producing reports, and wherein the metric M also depends on the number of distinct report scripts generated and saved.

43. The system of claim 29 wherein the enterprise information resource management system records metadata, and wherein the metric M also depends on the number of distinct metadata records.

44. A system for determining complexity of a metadata repository including a plurality of metadata constructs, comprising:

an input device for receiving a quantity of distinct metadata constructs, denoted by C;

a processor coupled to said input device for evaluating a metric of complexity, denoted by M, for a metadata repository having a capacity corresponding to C, according to a formula

$$M = f(C, X),$$

where f is a real-valued function of one or more real-valued parameters and X denotes optional additional parameters; and

a transaction processing system using the metric M for licensing the metadata repository.

45. The system of claim 44 wherein metadata constructs are instances of meta-model constructs.

46. The system of claim 45 wherein the meta-data constructs are constructs for meta-models of asset schemas, and wherein the price P also depends on the number of meta-model constructs.

47. The system of claim 45 wherein at least one meta-model is a schema for relational database schemas, and wherein at least one metadata construct corresponds to a table of a relational database schema.

48. The system of claim 45 wherein at least one meta-model is a schema for XML schemas, and wherein at least one metadata construct corresponds to a complex element of an XML schema.

49. A system for determining complexity of a metadata repository including a plurality of metadata constructs, the metadata constructs being instances of meta-model constructs for meta-models of schemas for data assets, comprising:

an input device for receiving a quantity of distinct meta-model constructs, denoted by C;

a processor coupled to said input device for evaluating a metric of complexity, denoted by M, for a metadata repository having a capacity corresponding to C, according to a formula

$$M = f(C, X),$$

where f is a real-valued function of one or more real-valued parameters and X denotes optional additional parameters; and

a transaction system receiving the metric M, for license of the metadata repository.

50. The system of claim 49 wherein at least one meta-model is a schema for relational database schemas, and wherein at least one metadata construct corresponds to a table of a relational database schema.

51. The system of claim 49 wherein at least one meta-model is a schema for XML schemas, and wherein at least one metadata construct corresponds to a complex element of an XML schema.

52. The system of claim 49 wherein the meta-model constructs include descriptors for data assets.

53. The system of claim 49 wherein the optional additional parameters include a parameter for a number of users of the repository.

54. The system of claim 49 wherein the optional additional parameters include a parameter for a number of available features.

55. The system of claim 54 wherein an available feature is the ability to change a meta-model.

56. The system of claim 54 wherein an available feature is the ability to batch scan metadata.

57. A computer-readable storage medium storing program code for causing a computer to determine complexity of an enterprise information resource management system, the enterprise information resource management system being used to contain an ontology into which a plurality of enterprise data assets are mapped, the ontology including a plurality of model constructs, the enterprise data assets including a plurality of assets constructs, and the mappings between the data assets and the ontology including a plurality of mapping constructs, by performing the steps of:

determining (i) a quantity of distinct asset constructs, denoted by C_{ASSET}, (ii) a quantity of distinct mapping constructs, denoted by $C_{MAPPING}$, and (iii) a quantity of distinct model constructs, denoted by C_{MODEL};

evaluating a metric of complexity, denoted by M, for an enterprise information resource management system having a capacity corresponding to C_{ASSET}, $C_{MAPPING}$ and C_{MODEL}, according to a formula

$$M = f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X),$$

where f is a real-valued function of three or more real-valued parameters and X denotes optional additional parameters; and

using the metric M within a transaction processing system, for license of the enterprise information resource management system.

58. A computer-readable storage medium storing program code for causing a computer to determine complexity of a metadata repository including a plurality of metadata constructs, by performing the steps of:

determining a quantity of distinct metadata constructs, denoted by C;

evaluating a metric of complexity, denoted by M, for a metadata repository having a capacity corresponding to C, according to a formula

$$M = f(C, X),$$

where f is a real-valued function of one or more real-valued parameters and X denotes optional additional parameters; and

using the metric M within a transaction processing system, for license of the metadata repository.

59. A computer-readable storage medium storing program code for causing a computer to determine complexity of a metadata repository including a plurality of metadata constructs, the metadata constructs being instances of meta-model constructs for meta-models of schemas for data assets, by performing the steps of:

determining a quantity of distinct meta-model constructs, denoted by C;

evaluating a metric of complexity, denoted by M, for a metadata repository having a capacity corresponding to C, according to a formula

$$M = f(C, X),$$

where f is a real-valued function of one or more real-valued parameters and X denotes optional additional parameters; and

using the metric M within a transaction processing system, for license of the metadata repository.